CLAIMS:

1. A method of enabling software development for an integrated circuit, the integrated circuit being configured to run a boot program that prevents unverified software from subsequently being loaded onto, or run by, the integrated circuit, the method including the step of loading an intermediate program onto the integrated circuit, the intermediate program being customised for a particular one or more of a plurality of potential integrated circuits that, when run on the processor, enables loading or running of code on only the particular one or more integrated circuits.

2. A method according to claim 1, wherein the intermediate program enables the loading or running of unverified code on only the particular one or more integrated circuits.

3. A method according claim 2, wherein the intermediate program enables the loading or running of the code only when the code includes data indicative of the particular one or more integrated circuits.

4. A method according claim 1, wherein the intermediate program includes an intermediate boot key, such that the intermediate program enables loading or running of the code only when the code is verified in accordance with the intermediate boot key.

5. An integrated circuit configured to run a boot program that prevents unverified software from subsequently being loaded onto, or run by, the integrated circuit.

6. An integrated circuit according to claim 5, programmed with program code configured to:
receive software data and a digital signature of the software data
generate a first digest from the software data; and
compare the first digest against a second digest obtained via the digital signature that accompanied the received software data;
wherein the program is considered valid when the first and second digests match.

7. An integrated circuit according to claim 6, wherein one or both of the digests were generated using a SHA1 function.

8. An integrated circuit according to claim 6, wherein the boot program contains a plurality of keys, and one of the keys is selected for use in generating the first digest, the key being selected in accordance with a selection criterion.

9. An integrated circuit according to claim 8, wherein the selection criterion is time-based, a particular one of the keys being selected depending on the time the selection is made.

10. An integrated circuit according to claim 8, wherein the selection criteria relates to a physical arrangement or configuration of the integrated circuit.

11. An integrated circuit according to claim 10, wherein the physical arrangement or configuration includes one or more of the following:

one or more pads wired to a reference voltage or to ground;

one or more fuses, one or more of which has been blown; or

the contents of non-volatile memory.

12. An integrated circuit according to claim 5, programmed with program code configured to:

receive encrypted software data,

decrypt the software data; and

validate the software data;

wherein the decrypted software is executed only when the validation is successful.

13. An integrated circuit according to claim 12, wherein the encryption function is RSA.

14. An integrated circuit according to claim 12, wherein the boot program contains a plurality of keys, and one of the keys is selected for use in decrypting the software data, the key being selected in accordance with a selection criterion.

15. An integrated circuit according to claim 14, wherein the selection criterion is time-based, a particular one of the keys being selected depending on the time the selection is made.

16. An integrated circuit according to claim 14, wherein the selection criteria relates to a physical arrangement or configuration of the integrated circuit.

17. An integrated circuit according to claim 16, wherein the physical arrangement or configuration includes one or more of the following:

one or more pads wired to a reference voltage or to ground;

one or more fuses, one or more of which has been blown; or

the contents of non-volatile memory.